

04-03-00

A

03/30/00  
JC777 U.S. PTO

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventorship.....Sierra et al.  
Applicant ..... Microsoft Corporation  
Attorney's Docket No. .... MS1-485US  
Title: Methods And Arrangements For Providing A Mark-Up Language Based Graphical User Interface For  
User Identification To An Operating System

TRANSMITTAL LETTER AND CERTIFICATE OF MAILING

To: Commissioner of Patents and Trademarks  
Washington, D.C. 20231  
From: Thomas A. Jolly (509) 324-9256  
Lee & Hayes, PLLC  
421 W. Riverside Avenue, Suite 500  
Spokane, WA 99201

JC530 U.S. PTO  
09/539231  
03/30/00

The following enumerated items accompany this transmittal letter and are being submitted for the matter identified in the above caption.

- 1. Transmittal Letter with Certificate of Mailing included.
- 2. PTO Return Postcard Receipt
- 3. Check in the Amount of \$1,048.00
- 4. Fee Transmittal
- 5. New patent application (title page plus 19 pages, including claims 1-29 & Abstract)
- 6. Executed Declaration
- 7. 3sheets of formal drawings (Figs. 1-3)
- 8. Assignment w/Recordation Cover Sheet

Large Entity Status [x] Small Entity Status [ ]

The Commissioner is hereby authorized to charge payment of fees or credit overpayments to Deposit Account No. 12-0769 in connection with any patent application filing fees under 37 CFR 1.16, and any processing fees under 37 CFR 1.17.

Date: 3-30-2000 By: Thomas A. Jolly  
Reg. No. 39,241

CERTIFICATE OF MAILING

I hereby certify that the items listed above as enclosed are being deposited with the U.S. Postal Service as either first class mail, or Express Mail if the blank for Express Mail No. is completed below, in an envelope addressed to The Commissioner of Patents and Trademarks, Washington, D.C. 20231, on the below-indicated date. Any Express Mail No. has also been marked on the listed items.

Express Mail No. (if applicable) EL580803665

Date: 3/30/00 By: Lori A. Vierra  
Lori A. Vierra

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**Methods And Arrangements For Providing A Mark-Up  
Language Based Graphical User Interface For User  
Identification To An Operating System**

Inventors:

Giampiero M. Sierra

&

Christopher A. Evans

ATTORNEY'S DOCKET NO. MS1-485US

1 **RELATED APPLICATION**

2 This application claims priority from U.S. Provisional Application Serial  
3 No. \_\_/\_\_\_\_\_, filed December 15, 1999 (Applicant's Docket Number  
4 149399.1, entitled "Web-Based User Interface For User Identification To The  
5 Operating System", express mailing label number EL425348720US), the  
6 disclosure of which is incorporated by reference herein.

7  
8 **TECHNICAL FIELD**

9 This invention relates to computers and software, and more particularly to  
10 methods and arrangements that provide a mark-up language based graphical user  
11 interface (GUI) that can be implemented to identify users to an operating system.

12  
13 **BACKGROUND OF THE INVENTION**

14 Computer systems are often protected by a logon program and/or other  
15 subsequent authentication programs that determine whether a user has permission  
16 to access certain computer system resources. By way of example, a networked  
17 computer may require that a user input a valid user name and password before the  
18 user is allowed to access network resources. Similarly, a Web site on the World  
19 Wide Web (WWW) portion of the Internet or on an intranet may require a valid  
20 user ID and password before the user is allowed to gain further access to various  
21 resources.

22 Controlling user access is not limited to networked computers. A single  
23 computer that is accessed by several users may also need to limit access to files  
24 and/or various programs therein. Thus, for example, in a home environment, a  
25 parent may decide to limit a child's access to the computer entirely, certain



## SUMMARY OF THE INVENTION

The present invention includes various methods and arrangements that can be implemented to identify users to an operating system through an advanced graphical user interface (GUI). The resulting GUI can be visually compelling and functional while advantageously remaining easy for the developer to create, maintain and modify.

Thus, for example, the above stated needs and others are met by a method that includes arranging for a markup language rendering engine to be loaded substantially near the beginning of an operating system initialization procedure, and providing markup language code suitable for use with the markup language rendering engine. The markup language is capable of soliciting at least one user input associated with a user logon process when rendered by the markup language rendering engine.

With the above example in mind, in accordance with certain implementations, a logon screen, for example, can be rendered from code written in Dynamic HTML (Hypertext Markup Language), eXtensible Markup Language (XML), eXtensible Hypertext Markup Language (XHTML), Standard Generalized Markup Language (SGML), or the like.

For the logon screen to be most effective upon initializing the computer, there will usually be a need to render the associated markup language file(s) early during the initialization stage. Accordingly, in certain implementations, the markup language rendering engine is loaded very near the beginning of the initialization of the operating system.

## **BRIEF DESCRIPTION OF THE DRAWINGS**

A more complete understanding of the various methods and arrangements of the present invention may be had by reference to the following detailed description when taken in conjunction with the accompanying drawings wherein:

Fig. 1 is a block diagram depicting an exemplary computer system.

Fig. 2 depicts an exemplary mark-up language based graphical user interface suitable for use in the computer system of Fig. 1 in identifying users to the operating system.

Fig. 3 is a flowchart depicting an exemplary process for identifying users to the operating system using a mark-up language based graphical user interface.

## **DETAILED DESCRIPTION**

As shown in Fig. 1, computer 20 includes one or more processors or processing units 21, a system memory 22, and a bus 23 that couples various system components including the system memory 22 to processors 21. Bus 23 represents one or more of any of several types of bus structures, including a memory bus or memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using any of a variety of bus architectures.

The system memory includes read only memory (ROM) 24 and random access memory (RAM) 25. A basic input/output system (BIOS) 26, containing the basic routines that help to transfer information between elements within computer 20, such as during start-up, is stored in ROM 24.

Computer 20 further includes a hard disk drive 27 for reading from and writing to a hard disk, not shown, a magnetic disk drive 28 for reading from and writing to a removable magnetic disk 29, and an optical disk drive 30 for reading

The drives and their associated computer-readable media provide nonvolatile storage of computer readable instructions, data structures, program modules and other data for computer 20. Although the exemplary environment described herein employs a hard disk, a removable magnetic disk 29 and a removable optical disk 31, it should be appreciated by those skilled in the art that other types of computer readable media which can store data that is accessible by a computer, such as magnetic cassettes, flash memory cards, digital video disks, random access memories (RAMs) read only memories (ROM), and the like, may also be used in the exemplary operating environment.

A number of program modules may be stored on the hard disk, magnetic disk 29, optical disk 31, ROM 24, or RAM 25, including an operating system 35, one or more application programs 36, other program modules 37, and program data 38. A user may enter commands and information into computer 20 through input devices such as keyboard 40 and pointing device 42. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are connected to the processing unit 21 through an interface 46 that is coupled to bus 23.

A monitor 47 or other type of display device is also connected to bus 23 via an interface, such as a video adapter 48. In addition to the monitor, personal computers typically include other peripheral output devices (not shown) such as speakers and printers.





Logon screen 100 can also be configured to display other user related information 116 to a user. In this example, user related information 116 identifies that Suzie has ten (10) new messages.

A second title area 106 is shown in this example along the left hand border of logon screen 100. Here, second title area 106 identifies the computer as the “Den Computer”. A selectable shut down mechanism 108 is also provided to allow a user to shut down the computer.

With the above example in mind, in accordance with certain implementations, logon screen 100 is a mark-up language based GUI. For example, a Dynamic HTML (Hypertext Markup Language) can be used to create logon screen 100. Dynamic HTML provides a mechanism to include a wide variety of functional as well as non-functional features to logon screen 100. Other types of mark-up languages and the like may also be used to define logon screen 100. For example, eXtensible Markup Language (XML), eXtensible Hypertext

For logon screen 100 to be effective upon initializing computer 20, there is a need to render the associated markup language file(s) early during the initialization stage. Accordingly, a markup language rendering engine (i.e., program) is loaded very near the beginning of the initialization of the operating system. Such rendering engines are well known. An exemplary markup language rendering engine is provided within Microsoft Internet Explorer (IE).

Dynamic HTML allows developers to create very attractive and colorful user interfaces. Thus, logon screen 100 may incorporate graphics and animations easily, while scripting complex behaviors, such as defining what happens when a user clicks on their name. The logon screen can be multi-layered and scaled to work with different resolutions. Additionally, complex graphical visual effects, such as, e.g., alpha blending, can be employed to create fades and transparencies that would be very difficult to implement in a traditional modal dialog interface.

By using a markup language and preloading the markup language rendering engine in the logon context, developers can advantageously prototype and generate dynamic user interfaces quickly and at a lower cost than would be required to significantly modify a conventional modal dialog. Thus, conceivably, each computer may have its own custom logon screen.

Having the ability to rapidly create and prototype logon screen designs will also make it easy for users to agree on what they like. For example, by changing the Cascading Style Sheets (CSS) and Dynamic HTML templates, developers can quickly change the logon screen without affecting the logon program code

In step 202, the logon program is initiated. This would occur upon rebooting computer 20, for example. Next, in step 204, a separate process, having a markup language rendering engine, is spawned to host the markup language content.

In step 206, the separate process retrieves user data from the operating system or elsewhere. The user data can include a listing of users, associated text identifiers 110, graphical identifiers 112, a password enabled identifier, and possibly, a password hint data (if enabled). Next, in step 208, the markup language rendering engine displays logon screen 100 along with applicable portions of the user data.

In step 210 the markup language rendering engine collects user inputs. This can include user mouse clicks, user typed text, audio commands, and/or other acceptable forms of user input. In the example of Fig. 2, William would select text identifier 110 or graphical identifier 112. Assuming that William has established a password (i.e., password is enabled), then he would need to enter his password.

Next, in step 212, the user inputs (e.g., user name and password) are provided to the logon program. In step 214, the logon program attempts to authenticate the user. If the user is authenticated, then a user desktop and/or workspace is created and subsequently displayed on display 47. If the user is not authenticated in step 214, then process 200 would return to either step 208, step 210, or otherwise handle the failed attempted logon.

Although some preferred embodiments of the various methods and arrangements of the present invention have been illustrated in the accompanying Drawings and described in the foregoing Detailed Description, it will be understood that the invention is not limited to the exemplary embodiments disclosed, but is capable of numerous rearrangements, modifications and substitutions without departing from the spirit of the invention as set forth and defined by the following claims.

1 **CLAIMS**

2 What is claimed is:

3  
4 1. A method comprising:  
5 arranging for a markup language rendering engine to be loaded  
6 substantially near the beginning of an operating system initialization procedure;  
7 and

8 providing markup language code suitable for use with the markup language  
9 rendering engine, the markup language being capable of soliciting at least one user  
10 input when rendered by the markup language rendering engine, the user input  
11 being associated with a user logon process.

12  
13 2. The method as recited in Claim 1, wherein providing the markup  
14 language code further includes providing user data, the user data being operatively  
15 associated with the user logon process.

16  
17 3. The method as recited in Claim 2, wherein the user data includes  
18 data selected from a set comprising a list of users, a text identifier, a graphical  
19 identifier, a password enabled identifier, and password hint data, and related user  
20 information data.

21  
22 4. The method as recited in Claim 2, further comprising:  
23 configuring the markup language rendering engine to display at least a  
24 portion of the user data based on the markup language code.  
25

1           5.     The method as recited in Claim 1, further comprising:  
2           configuring the markup language code to provide the user input to an  
3           authorization entity for validation determination.

4  
5           6.     The method as recited in Claim 1, wherein the user input includes at  
6           least one input selected from a group of inputs comprising a user name, a user  
7           identifier, and a password.

8  
9           7.     The method as recited in Claim 1, wherein the markup language  
10          code includes markup language code selected from at least one markup language  
11          in a group comprising hypertext markup language (HTML), Dynamic Hypertext  
12          Markup Language (DHTML), eXtensible Markup Language (XML), eXtensible  
13          Hypertext Markup Language (XHTML), and Standard Generalized Markup  
14          Language (SGML).

15  
16          8.     A computer-readable medium having computer-executable  
17          instructions for performing steps comprising:

18                 arranging for a markup language rendering engine to be loaded  
19                 substantially near the beginning of an operating system initialization procedure;  
20                 and

21                 providing markup language code suitable for use with the markup language  
22                 rendering engine, the markup language being capable of soliciting at least one user  
23                 input when rendered by the markup language rendering engine, the user input  
24                 being associated with a user logon process.



markup language in a group comprising hypertext markup language (HTML), Dynamic Hypertext Markup Language (DHTML), eXtensible Markup Language (XML), eXtensible Hypertext Markup Language (XHTML), and Standard Generalized Markup Language (SGML).

15. An arrangement including a memory, a data storage device, a display device, and a processor operatively coupled to the memory, data storage device and the display device, the arrangement comprising:

a markup language rendering engine stored within the data storage device and suitable for loading in the memory substantially near the beginning of an operating system initialization procedure; and

markup language code suitable stored in the data storage device and configurable for use with the markup language rendering engine, the markup language being capable of soliciting at least one user input when rendered by the markup language rendering engine onto the display device, the user input being associated with a user logon process.

16. The arrangement as recited in Claim 15, further comprising user data stored in the data storage device and configurable for use with the markup language rendering engine, the user data being operatively associated with the user logon process.

17. The arrangement as recited in Claim 16, wherein the user data includes data selected from a set comprising a list of users, a text identifier, a



graphical identifier, a password enabled identifier, and password hint data, and related user information data.

18. The arrangement as recited in Claim 16, wherein the markup language rendering engine is further configurable to display at least a portion of the user data on the display device based on the markup language code.

19. The arrangement as recited in Claim 15, further comprising an authorization entity stored within the data storage device, and wherein the markup language rendering engine is further configurable to provide the user input to the authorization entity for validation determination based on the markup language code.

20. The arrangement as recited in Claim 15, wherein the user input includes at least one input selected from a group of inputs comprising a user name, a user identifier, and a password.

21. The arrangement as recited in Claim 15, wherein the markup language code includes markup language code selected from at least one markup language in a group comprising hypertext markup language (HTML), Dynamic Hypertext Markup Language (DHTML), eXtensible Markup Language (XML), eXtensible Hypertext Markup Language (XHTML), and Standard Generalized Markup Language (SGML).



26. The method as recited in Claim 22, wherein the markup language code includes markup language code selected from at least one markup language in a group comprising hypertext markup language (HTML), Dynamic Hypertext Markup Language (DHTML), eXtensible Markup Language (XML), eXtensible Hypertext Markup Language (XHTML), and Standard Generalized Markup Language (SGML).

27. A markup language based logon user interface arrangement for use in logging users onto an operating system of a computer, the user interface comprising:

a logon screen;

a user logon area within the logon screen, the user logon area visually identifying a plurality of users using text identifiers and graphical identifiers, such that each text identifier and graphical identifier are selectable by the user through the user interface and upon selection by the user cause the user interface to prompt the user to input a password; and

a single selectable shut down mechanism graphically located within the logon screen and configured to shut the computer down when selected through the user interface by the user.

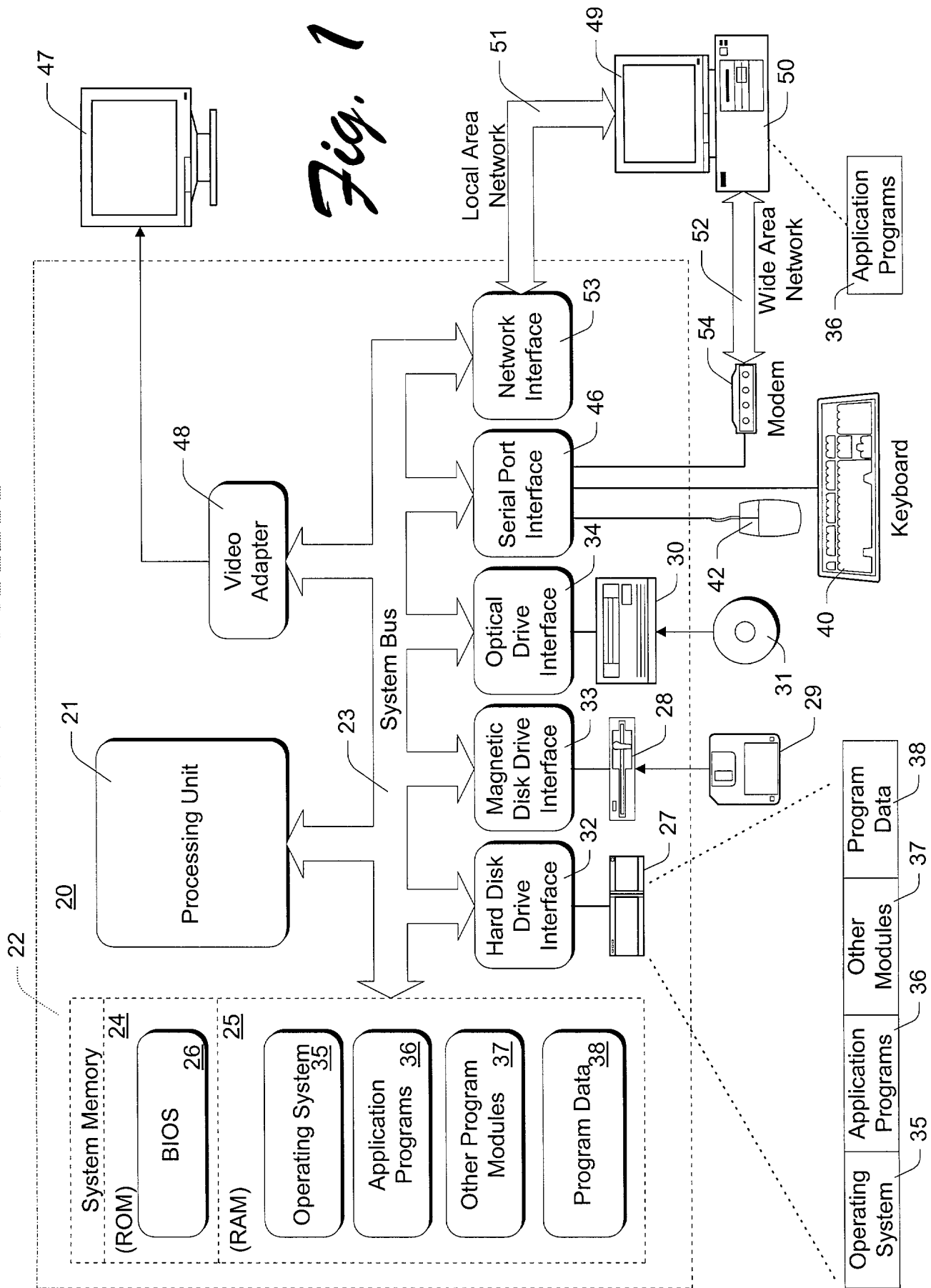
28. The user interface as recited in Claim 27, wherein the logon screen is rendered substantially near the beginning of the initialization of the operating system using a markup language rendering engine.



[illegible]

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25

Fig. 1



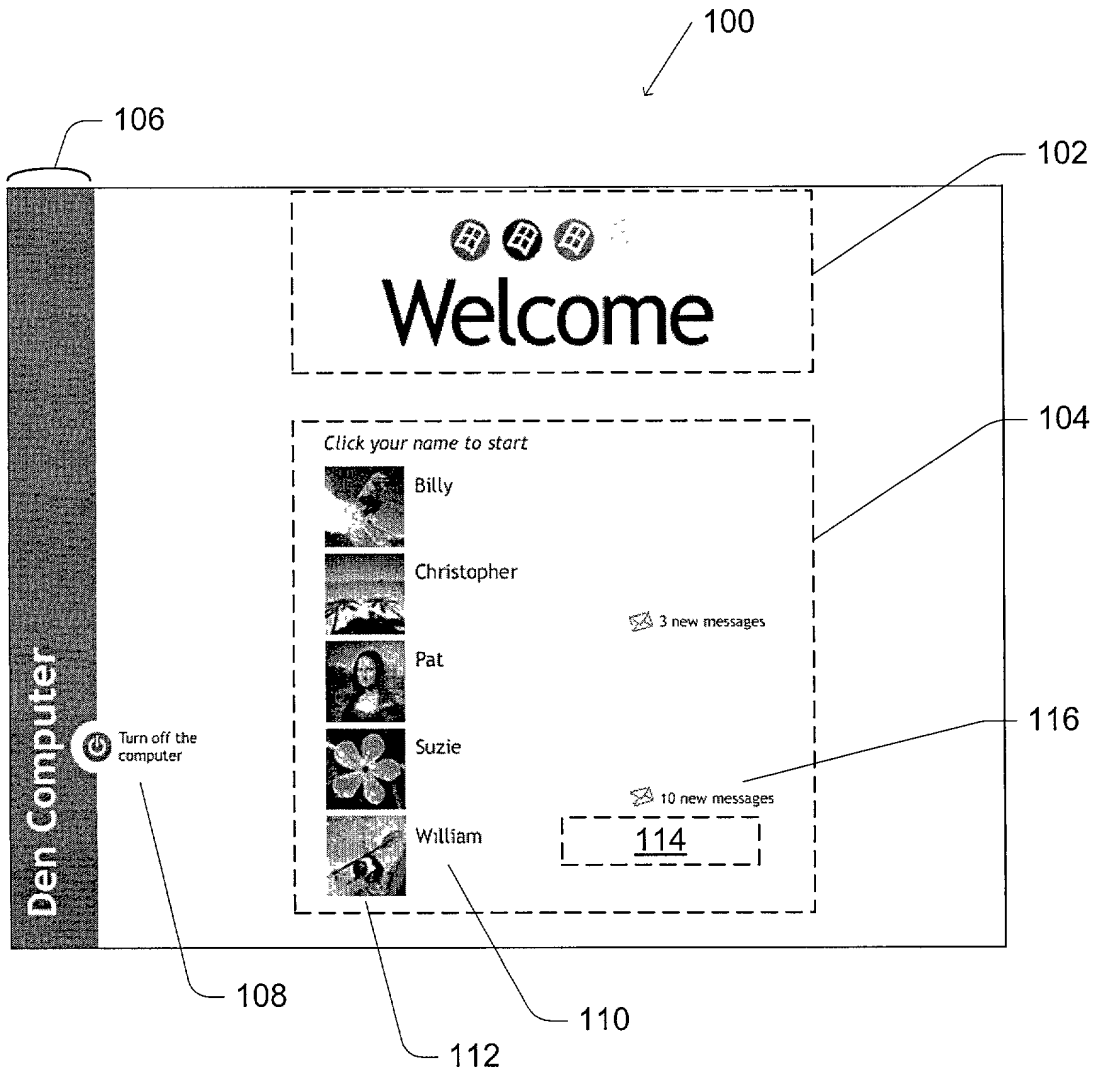
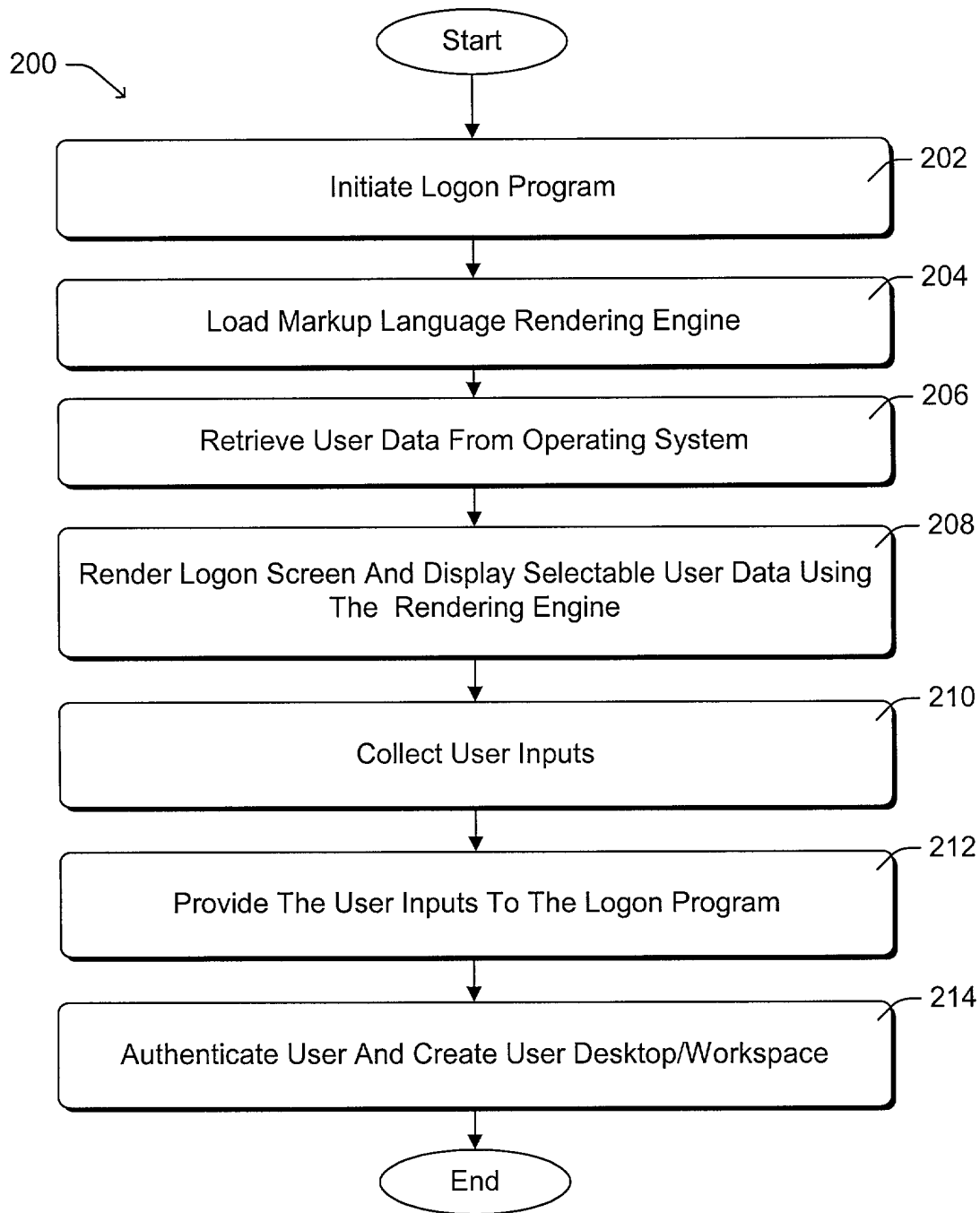


Fig. 2



*Fig. 3*



1       **IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

2   Inventorship.....Sierra et al.  
 3   Applicant ..... Microsoft Corporation  
 4   Attorney's Docket No. .... MS1-485US  
 Title: Methods And Arrangements For Providing A Mark-Up Language Based  
 Graphical User Interface For User Identification To An Operating System

5       **DECLARATION FOR PATENT APPLICATION**

6       As a below named inventor, I hereby declare that:

7       My residence, post office address and citizenship are as stated below next to  
 8   my name.

9       I believe I am the original, first and sole inventor (if only one name is listed  
 10   below) or an original, first and joint inventor (if plural names are listed below) of the  
 11   subject matter which is claimed and for which a patent is sought on the invention  
 12   entitled "Methods And Arrangements For Providing A Mark-Up Language Based  
 13   Graphical User Interface For User Identification To An Operating System," the  
 14   specification of which is attached hereto.

15       I have reviewed and understand the content of the above-identified  
 16   specification, including the claims.

17       I hereby claim benefit under 35 U.S.C. 119(e) of United States Provisional  
 18   Application \_\_/\_\_\_\_\_, filed December 15, 1999 (Applicant's Docket Number  
 19   149399.1, entitled "Web-Based User Interface For User Identification To The  
 20   Operating System").

21       I acknowledge the duty to disclose information which is material to the  
 22   examination of this application in accordance with Title 37, Code of Federal  
 23   Regulations, § 1.56(a).

1 PRIOR FOREIGN APPLICATIONS: no applications for foreign patents or  
2 inventor's certificates have been filed prior to the date of execution of this  
3 declaration.

4 **Power of Attorney**

5 I appoint the following attorneys to prosecute this application and transact all  
6 future business in the Patent and Trademark Office connected with this application:  
7 Lewis C. Lee, Reg. No. 34,656; Daniel L. Hayes, Reg. No. 34,618; Allan T.  
8 Sponseller, Reg. 38,318; Steven R. Sponseller, Reg. No. 39,384; James R.  
9 Banowsky, Reg. No. 37,773; Lance R. Sadler, Reg. No. 38,605; Michael A. Proksch,  
10 Reg. No. 43,021; Thomas A. Jolly, Reg. No. 39,241; David A. Morasch, Reg. No.  
11 42,905; Kasey C. Christie, Reg. No. 40,559; Katie E. Sako, Reg. No. 32,628 and  
12 Daniel D. Crouse, Reg. No. 32,022.

13 Send correspondence to: LEE & HAYES, PLLC, 421 W. Riverside Avenue,  
14 Suite 500, Spokane, Washington, 99201. Direct telephone calls to: Thomas A. Jolly  
15 (509) 324-9256.

16  
17 All statements made herein of my own knowledge are true and that all  
18 statements made on information and belief are believed to be true; and further that  
19 these statements were made with the knowledge that willful false statements and the  
20 like so made are punishable by fine or imprisonment, or both, under Section 1001 of  
21 Title 18 of the United States Code and that such willful false statement may  
22 jeopardize the validity of the application or any patent issued therefrom.  
23  
24  
25

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25

Post Office Address: 2815 Boylston Ave E., Apt. 406  
Seattle, WA 98012

Post Office Address: 21002 NE 44th Street  
Redmond, WA 98053